Prompt: crea un conjunto de endpoints para cada una de estas funciones

## Endpoints básicos de la API

Recuerda trabajar sobre el archivo `main.py`

1. ### Bubble Sort

- Ruta: `/bubble-sort`

- Método: `POST`

- Descripción: Recibe una lista de números y devuelve la lista ordenada utilizando el algoritmo de Bubble Sort.

- Entrada: `{"numbers": [lista de números]}`

- Salida: `{"numbers": [lista de números ordenada]}`

2. ### Filtro de Pares

- Ruta: `/filter-even`

- Método: `POST`

- Descripción: Recibe una lista de números y devuelve únicamente aquellos que son pares.

- Entrada: `{"numbers": [lista de números]}`

- Salida: `{"even\_numbers": [lista de números pares]}`

3. ### Suma de Elementos

- Ruta: `/sum-elements`

- Método: `POST`

- Descripción: Recibe una lista de números y devuelve la suma de sus elementos.

- Entrada: `{"numbers": [lista de números]}`

- Salida: `{"sum": suma de los números}`

4. ### Máximo Valor

- Ruta: `/max-value`

- Método: `POST`

- Descripción: Recibe una lista de números y devuelve el valor máximo.

- Entrada: `{"numbers": [lista de números]}`

- Salida: `{"max": número máximo}`

5. ### Búsqueda Binaria

- Ruta: `/binary-search`

- Método: `POST`

- Descripción: Recibe un número y una lista de números ordenados. Devuelve true y el índice si el número está en la lista, de lo contrario false y -1 como index.

- Entrada: `{"numbers": [lista de números], "target": int}`

- Salida: `{"found": booleano, "index": int}`

### Nota

El payload debe estar definido por:

```python

class Payload(BaseModel):

numbers: List[int]

```

o

```python

class BinarySearchPayload(BaseModel):

numbers: List[int]

target: int

```

RESPUESTA

To create the endpoints as described, you need to define the routes and implement the logic for each function in your FastAPI application. Below is the implementation for each endpoint in the `main.py` file.

\*\*File: /home/gandrade/Descargas/CAP01\_CHALLENGE\_GENITO/main.py\*\*

```python

from fastapi import FastAPI, HTTPException

from typing import List, Any

from pydantic import BaseModel

from passlib.context import CryptContext

import jwt

fake\_db = {"users": {}}

app = FastAPI()

class Payload(BaseModel):

numbers: List[int]

class BinarySearchPayload(BaseModel):

numbers: List[int]

target: int

@app.post("/bubble-sort")

def bubble\_sort(payload: Payload):

numbers = payload.numbers

n = len(numbers)

for i in range(n):

for j in range(0, n-i-1):

if numbers[j] > numbers[j+1]:

numbers[j], numbers[j+1] = numbers[j+1], numbers[j]

return {"numbers": numbers}

@app.post("/filter-even")

def filter\_even(payload: Payload):

numbers = payload.numbers

even\_numbers = [num for num in numbers if num % 2 == 0]

return {"even\_numbers": even\_numbers}

@app.post("/sum-elements")

def sum\_elements(payload: Payload):

numbers = payload.numbers

total\_sum = sum(numbers)

return {"sum": total\_sum}

@app.post("/max-value")

def max\_value(payload: Payload):

numbers = payload.numbers

if not numbers:

raise HTTPException(status\_code=400, detail="List is empty")

max\_num = max(numbers)

return {"max": max\_num}

@app.post("/binary-search")

def binary\_search(payload: BinarySearchPayload):

numbers = payload.numbers

target = payload.target

left, right = 0, len(numbers) - 1

while left <= right:

mid = (left + right) // 2

if numbers[mid] == target:

return {"found": True, "index": mid}

elif numbers[mid] < target:

left = mid + 1

else:

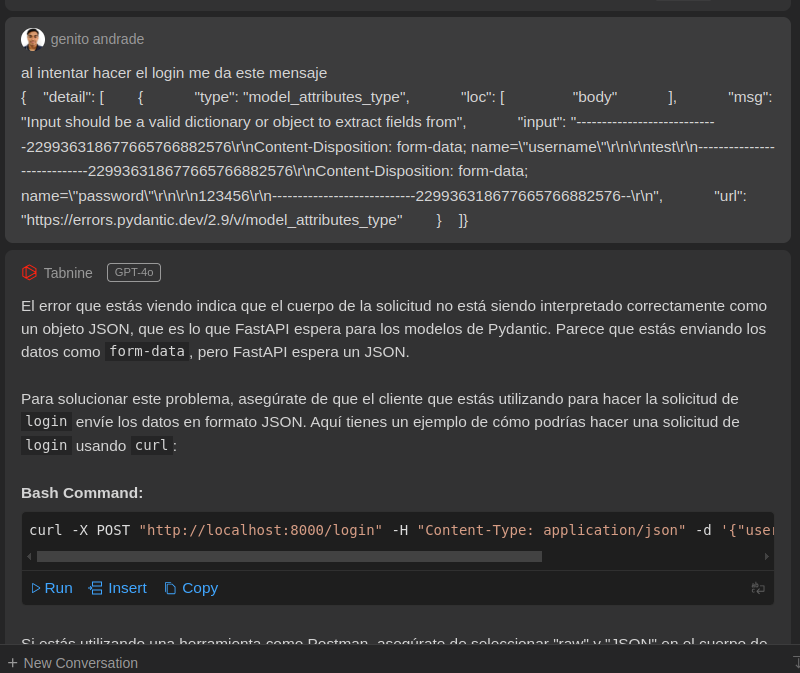
right = mid - 1

return {"found": False, "index": -1}

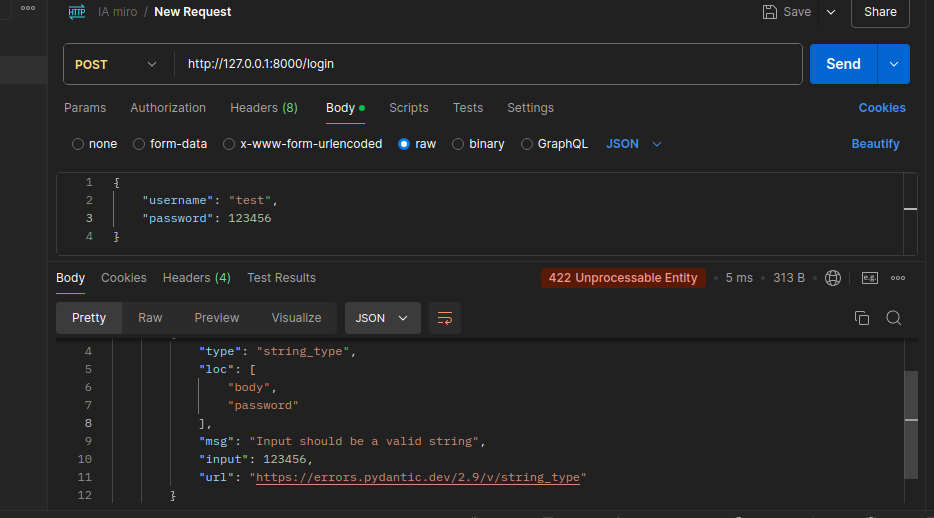
```

This code defines the endpoints for bubble sort, filtering even numbers, summing elements, finding the maximum value, and performing a binary search. Each endpoint receives a JSON payload, processes the data, and returns the appropriate response.

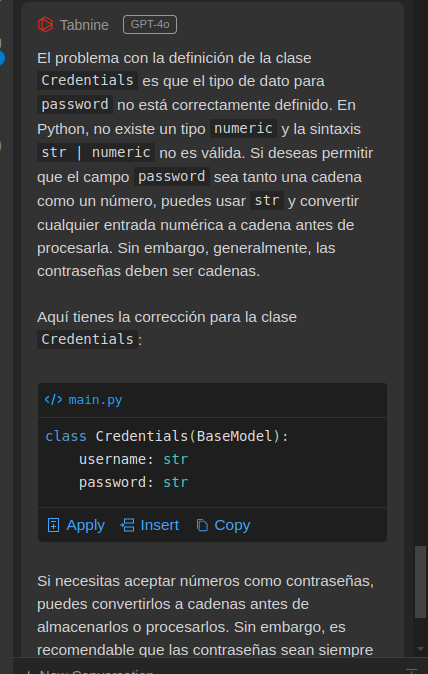
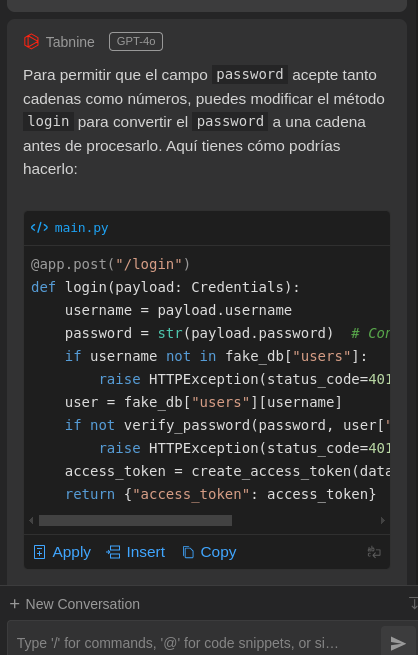
Primer intento de login a modo de pruebas



segundo intento de login

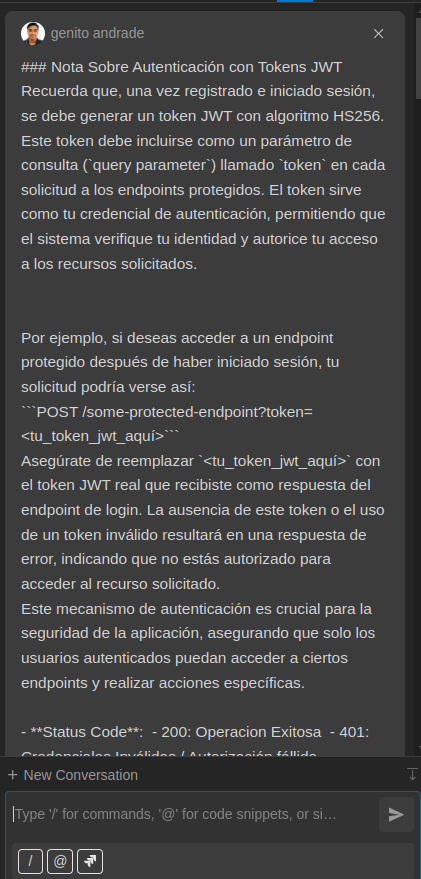


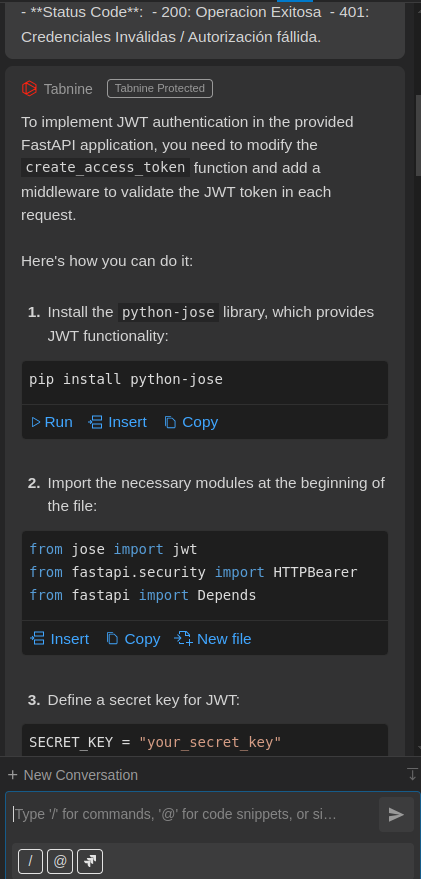
Correccion para recibir password como numero

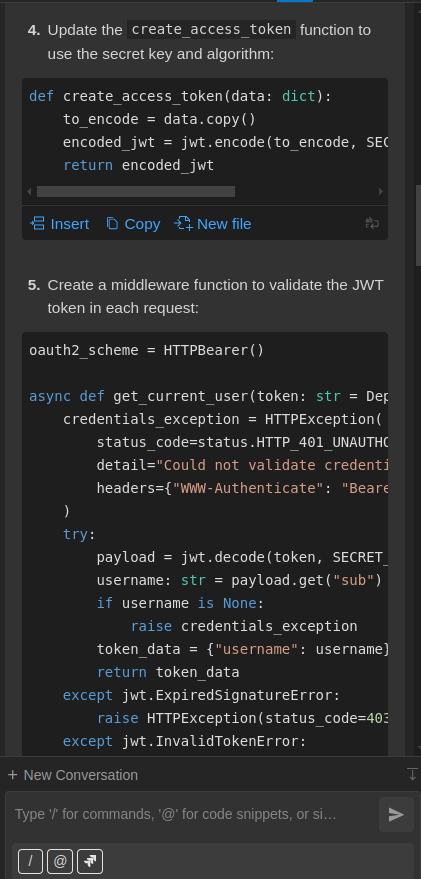
 

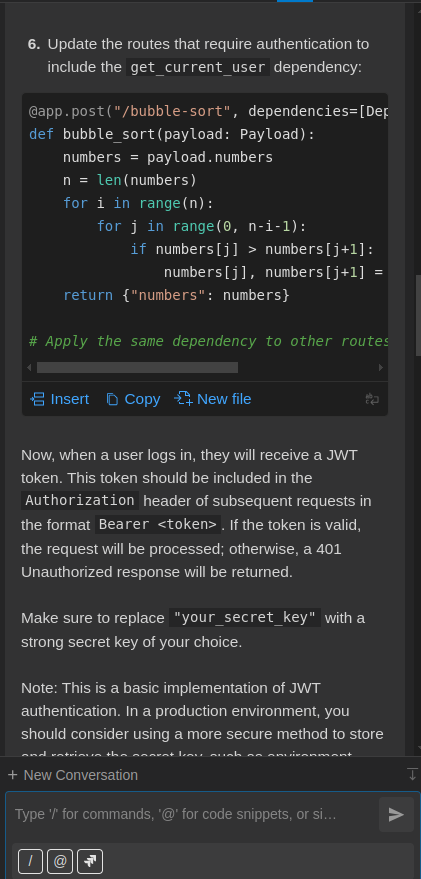
Consultar los usurios registrados

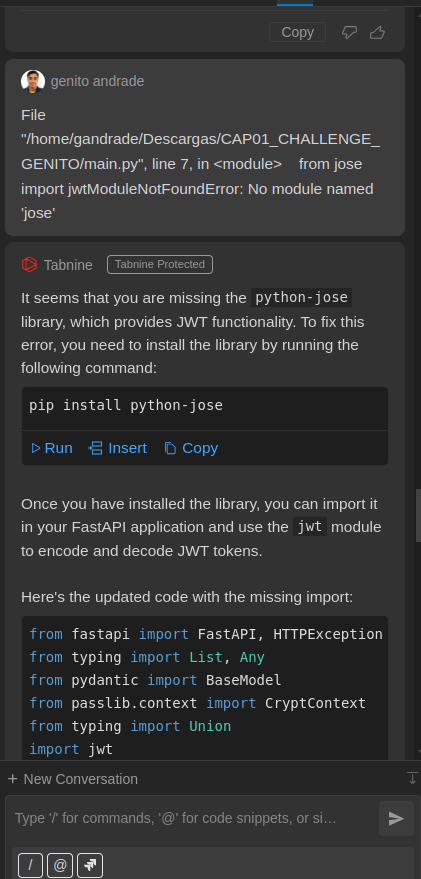












Documentacion de codigo

